WHAT IS CLAIMED IS:

1. A method for providing interface compatibility between two hierarchical collections of integrated circuit (IC) design objects, comprising:

establishing an associative correspondence between a design object from a first hierarchical collection and a design object from a second hierarchical collection;

generating a port compatibility map based on determination that a particular associative correspondence includes a pair of design objects, one from each hierarchical collection, that are port-compatible; and

reducing said port compatibility map to determine a set of design object pairs that allow replaceability between said first and second hierarchical collections.

2. The method for providing interface compatibility between two hierarchical collections of IC design objects as recited in claim 1, wherein said operation of reducing said port compatibility map is performed iteratively.

3. The method for providing interface compatibility between two hierarchical collections of IC design objects as recited in claim 2, wherein said operation of reducing said port compatibility map further comprises:

pruning design object entities of said port compatibility map based on determination that a specific design object at a particular hierarchical level is exchangeable between said first and second hierarchical collections regardless of whether any design objects respectively under said specific design objects are port-incompatible, thereby generating a compatibility map derived from said port compatibility map;

for each design object of said compatibility map whose Boolean value is false, determining if a user design object thereof can be falsified, wherein said user design object is a parent design object in said compatibility map whose Boolean value is true;

if so, rendering said user design object's Boolean value to be false; and

falsifying remaining design objects of said compatibility map until all parent design objects therein are exhausted, thereby arriving at a set of design objects whose Boolean values indicate whether they can be replaced between said first and second hierarchical collections.

4. The method for providing interface compatibility between two hierarchical collections of IC design objects as recited in claim 3, wherein said determination that said pair of design objects are port-compatible is effectuated automatically by comparing signal names of said design objects.

5. The method for providing interface compatibility between two hierarchical collections of IC design objects as recited in claim 3, wherein said determination that said pair of design objects are port-compatible is effectuated by establishing semantic equivalency between said design objects.

6. The method for providing interface compatibility between two hierarchical collections of IC design objects as recited in claim 3, wherein said determination that said pair of design objects are port-compatible is effectuated comparing input/output (I/O) interfaces of said design objects.

7. The method for providing interface compatibility between two hierarchical collections of IC design objects as recited in claim 3, wherein said determination that said pair of design objects are port-compatible is effectuated manually.

8. The method for providing interface compatibility between two hierarchical collections of IC design objects as recited in claim 3, wherein said determination that said pair of design objects are port-compatible is effectuated by enforcing electrical design equivalency between said design objects.

9. A system for providing interface compatibility between two hierarchical collections of integrated circuit (IC) design objects, comprising:

means for establishing an associative correspondence between a design object from a first hierarchical collection and a design object from a second hierarchical collection;

means for generating a port compatibility map based on determination that a particular associative correspondence includes a pair of design objects, one from each hierarchical collection, that are port-compatible; and

means for reducing said port compatibility map to determine a set of design object pairs that allow replaceability between said first and second hierarchical collections.

10. The system for providing interface compatibility between two hierarchical collections of IC design objects as recited in claim 9, wherein said means for reducing said port compatibility map is operable to perform iteratively.

11. The system for providing interface compatibility between two hierarchical collections of IC design objects as recited in claim 10, wherein said means for reducing said port compatibility map further comprises:

means for pruning design object entities of said port compatibility map based on determination that a specific design object at a particular hierarchical level is exchangeable between said first and second hierarchical collections regardless of whether any design objects respectively under said specific design objects are port-incompatible, thereby generating a compatibility map derived from said port compatibility map;

means for determining, for each design object of said compatibility map whose Boolean value is false, if a user design object thereof can be falsified, wherein said user design object is a parent design object in said compatibility map whose Boolean value is true;

means, responsive to said determination that a design object of said compatibility map can be falsified, for rendering said user design object's Boolean value to be false; and

means for falsifying remaining design objects of said compatibility map until all parent design objects therein are exhausted, thereby arriving at a set of design objects whose Boolean values indicate whether they can be replaced between said first and second hierarchical collections.

12. The system for providing interface compatibility between two hierarchical collections of IC design objects as recited in claim 11, wherein said determination that said pair of design objects are port-compatible is effectuated automatically by comparing signal names of said design objects.

13. The system for providing interface compatibility between two hierarchical collections of IC design objects as recited in claim 11, wherein said determination that said pair of design objects are port-compatible is effectuated by establishing semantic equivalency between said design objects.

14. The system for providing interface compatibility between two hierarchical collections of IC design objects as recited in claim 11, wherein said determination that said pair of design objects are port-compatible is effectuated comparing input/output (I/O) interfaces of said design objects.

15. The system for providing interface compatibility between two hierarchical collections of IC design objects as recited in claim 11, wherein said determination that said pair of design objects are port-compatible is effectuated manually.

16. The system for providing interface compatibility between two hierarchical collections of IC design objects as recited in claim 11, wherein said determination that said pair of design objects are port-compatible is effectuated by enforcing electrical design equivalency between said design objects.

17. A computer platform operable to support an integrated circuit (IC) chip design database environment, comprising:

a first collection of design objects disposed in said IC chip design database environment;

a second collection of design objects disposed in said IC chip design database environment;

a port compatibility engine operating to generate a port-compatible group of design objects relative to said first and second collections of design objects;

a compatibility engine for obtaining a compatible group of design objects based on said port-compatible group; and

a reduction engine for iteratively reducing said compatibility group to a set of design objects that are replaceable between said first and second collections of design objects.

18. The computer platform operable to support an IC chip design database environment as recited in claim 17, wherein said first collection of design objects is hierarchically arranged.

19. The computer platform operable to support an IC chip design database environment as recited in claim 17, wherein said second collection of design objects is hierarchically arranged.

20. The computer platform operable to support an IC chip design database environment as recited in claim 17, wherein each of said first and second collections of design objects comprises a standard cell library portion.

21. A computer-accessible medium operable with a computer platform to support an integrated chip (IC) chip design database environment, the medium having stored thereon instructions for providing interface compatibility between two hierarchical collections of IC design objects, comprising:

program code for establishing an associative correspondence between a design object from a first hierarchical collection and a design object from a second hierarchical collection;

program code for generating a port compatibility map based on determination that a particular associative correspondence includes a pair of design objects, one from each hierarchical collection, that are port-compatible; and

program code for reducing said port compatibility map to determine a set of design object pairs that allow replaceability between said first and second hierarchical collections.

22. The computer-accessible medium as recited in claim 21, wherein said program code for reducing said port compatibility map is operable to perform iteratively.

23. The computer-accessible medium as recited in claim 22, wherein said program code for reducing said port compatibility map further comprises:

program code for pruning design object entities of said port compatibility map based on determination that a specific design object at a particular hierarchical level is exchangeable between said first and second hierarchical collections regardless of whether any design objects respectively under said specific design objects are port-incompatible, thereby generating a compatibility map derived from said port compatibility map;

program code for determining, for each design object of said compatibility map whose Boolean value is false, if a user design object thereof can be falsified, wherein said user design object is a parent design object in said compatibility map whose Boolean value is true;

program code, responsive to said determination that a design object of said compatibility map can be falsified, for rendering said user design object's Boolean value to be false; and

program code for falsifying remaining design objects of said compatibility map until all parent design objects therein are exhausted, thereby arriving at a set of design objects whose Boolean values indicate whether they can be replaced between said first and second hierarchical collections.

24. The computer-accessible medium as recited in claim 23, wherein said determination that said pair of design objects are port-compatible is effectuated automatically by comparing signal names of said design objects.

25. The computer-accessible medium as recited in claim 23, wherein said determination that said pair of design objects are port-compatible is effectuated by establishing semantic equivalency between said design objects.

26. The computer-accessible medium as recited in claim 23, wherein said determination that said pair of design objects are port-compatible is effectuated comparing input/output (I/O) interfaces of said design objects.

27. The computer-accessible medium as recited in claim 23, wherein said determination that said pair of design objects are port-compatible is effectuated manually.

28. The computer-accessible medium as recited in claim 23, wherein said determination that said pair of design objects are port-compatible is effectuated by enforcing electrical design equivalency between said design objects.